

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Absolvování individuální odborné praxe
Individual Professional Practise in the
Company

2009

Jan Šaržík

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 7. Května 2009

.....

Abstrakt

Cílem této bakalářské práce, je seznámení s mou praxí ve firmě TietoEnator, která se zabývá širokým spektrem vývoje softwaru. Seznámení s profesionálním vývojem J2ME aplikací pro S60 platformy. Při implementaci je kladen důraz na jednoduchost a použitelnost aplikací pro budoucí uživatele. Implementace je založena na open-source technologiích Java a J2ME.

Klíčové slova

S60 platforma, J2ME, Java

Abstract

The purpose of this bachelor thesis is an introduction with my practice in TietoEnator company, which is involved in a broad spectrum of software development activities. The Introduction of the professional development of J2ME applications for S60 platform During the implementation the simplicity and usability for potential users is emphasized. Implementation is based on Java and J2ME open-source technologies.

Keywords

S60 platform, J2ME, Java

Seznam použitých symbolů a zkratek

AMS	- application managment software
API	- Application Programming Interface
CLDC	- Connected Limited Device Configuration
HTML	- HyperText Markup Language
IDE	- Integrated development environment
J2ME	- Java 2 Micro Edition
MIDP	- Mobile Information Device Profile
MMAPI	- Mobile media Application Programming Interface
PHP	- Hypertext Preprocesor
UML	- Unified Modeling Language
USB	- Universal Serial Bus
WTK	- Wireless Toolkit
XML	- Extensible Markup Language

Obsah

1 Úvod.....	4
2 Odborné zaměření firmy TietoEnator	5
3 Seznámení se základními knihovnami, třídami a prostředky užitých při vývoji J2ME aplikací	6
3.1 Základní knihovny	6
3.1.1 MIDlet.....	6
3.1.2 CLDC - Connected Limited Device Configuration	6
3.1.3 MIDP – Mobile Information Device Profile	6
3.1.4 MMAPI – Mobile media API	6
3.2 Základní třídy	6
3.2.1 javax.microedition.midlet	6
3.2.2 javax.microedition.lcdui.Display	6
3.2.3 javax.microedition.lcdui.Displayable.....	7
3.2.4 javax.microedition.lcdui.CommandListener	7
3.2.5 javax.microedition.lcdui.Screen.....	7
3.2.6 javax.microedition.lcdui.Form	7
3.2.7 Canvas	7
3.2.8 Manager	7
3.2.9 Player	7
3.3 Prostředky potřebné pro vývoj aplikací	7
3.3.1 Vývojové prostředí.....	7
3.3.2 Emulátory	8
3.3.3 Mobilní telefony	8
4 Zadáání úkolů	9
4.1 Konfigurace vývojového prostředí Eclipse pro J2ME	9
4.2 První Midlet pro seznámení s J2ME	9
4.3 Vytvořte Midlet MMAPIPlayer pro přehrávání všech podporujících multimediálních souborů.....	9
4.4 Rozšíření Midletu MMAPIPlayer o další funkčnost.....	9
4.5 Podepsání Midletu MMAPIPlayer	9
5 Řešení zadaných úkolů.....	10

OBSAH	2
5.1 Konfigurace vývojového prostředí.....	10
5.1.1 Konfigurace Eclipse	10
5.1.2 Konfigurace Sun Wireless Toolkit.....	10
5.1.3 Emulátor NOKIA.....	10
5.2 Vytvoření prvního Midletu	11
5.2.1 Seznámení se s principem fungování Midletu	11
5.2.2 Rozhodnutí o použití komponent pro základní funkce Midletu.....	11
5.2.3 Vlastní řešení prvního projektu.....	12
5.2.4 Zabalení, instalace a spuštění Midletu	12
5.2.5 Obrazovky midletu.....	13
5.3 Vytvoření midletu MMAPi.....	13
5.3.1 Plánování.....	13
5.3.2 Zvolené komponenty pro MMAPi midlet.....	14
5.3.3 Vlastní řešení MMAPi midletu	14
5.3.4 Odzkoušení a předání Midletu	15
5.4 Rozšíření midletu MMAPi.....	15
5.4.1 Plánování.....	15
5.4.2 Zvolené komponenty pro rozšíření midletu	15
5.4.3 Vlastní rozšíření midletu MMAPi	15
5.4.4 Odzkoušení a předání midletu.....	16
5.5 Podepsání midletu	16
5.5.1 Podepsání midletu TotalCommander	16
5.5.2 Odzkoušení midletu	16
6 Závěr	17
7. Literatura.....	18
1. Beginning J2ME Platform.....	18
2. J2ME (online)	18

Seznam Obrázků

Obrázek 1: Mobilní telefon s dotykovým displejem	6
Obrázek 2: Mobilní telefon bez dotykového displeje	6
Obrázek 3: Život midletu	9
Obrázek 4 Komponenty zobrazitelné na displeji	9
Obrázek 5 Obrazovky prvního midletu:	11

1 Úvod

Profesionální vývoj aplikací určených pro telekomunikační technologie je jednou z nejdůležitějších součástí nejen pro mobilní telefony, ale také pro vše s telekomunikačními zařízeními spojené. Aplikace pro tyto zařízení jsou používány dnes a denně, aniž si to dnešní konzumní společnost vůbec neuvědomuje.

Tato bakalářská práce přiblíží čtenáři problematiku jednoduchého profesionálního vývoje aplikací užitých při práci na S60 platformách. Tyto aplikace zde nejsou užity pro koncového zákazníka jako takového, ale pro ulehčení práce profesionálního programátora pracujícího na platformách mobilních telefonů. Vytvořená aplikace umožní programátorovi snadnou údržbu platformy, na které právě pracuje.

V druhé kapitole této práce je čtenáři přiblíženo odborné zaměření firmy TietoEnator známou nejen v České Republice ale i po celém světě. Také zde jsou přiblíženy různé průmyslové odvětví, ve kterých firma působí.

Třetí kapitola seznámí čtenáře se základními knihovnami, třídami a potřebnými prostředky, užitých při vývoji aplikací pro mobilní telefony.

Ve čtvrté kapitole je ukázáno, jak může vypadat jednoduché zadání několika jednoduchých J2ME aplikací pro mobilní telefony.

Pátá kapitola zprostředkovává řešení jednoduchých úkolů ze čtvrté kapitoly.

V závěru je probráno krátké shrnutí dosažených poznatků a dovedností získaných při konání zadaných úkolů.

2 Odborné zaměření firmy TietoEnator

Tieto je společnost poskytující služby v oblasti IT, výzkumu, vývoje a poradenství. Se svými 16 000 odborníky je jedním z největších poskytovatelů IT služeb v severní Evropě a je přední světovou společností ve vybraných odvětvích.

Zaměřuje se na oblasti, ve kterých má ty nejdůkladnější znalosti podnikání a potřeb jejich zákazníků. Její stoprocentní zaměření na potřeby zákazníka a odborné znalosti ze Skandinávských zemí jí odlišují od jejich konkurentů.

Na jejich hlavních trzích - severní Evropa, Německo a Rusko - se zaměřuje na poskytování služeb velkým a středně velkým organizacím. Celosvětově spolupracuje se svými zákazníky v oblasti telekomunikací a digitálních služeb a v lesním, ropném a plynárenském průmyslu. Úzce spolupracuje s předními světovými společnostmi a organizacemi a roste společně s nimi.

Od 1. prosince 2008 používá novou obchodní značku Tieto. Akcie Tieto jsou obchodovány na burzách NASDAQ OMX v Helsinkách a ve Stockholmu.

V této firmě jsem byl zařazen do oddělení Devices Research & Development.

3 Seznámení se základními knihovnami, třídami a prostředky užitých při vývoji J2ME aplikací

3.1 Základní knihovny

3.1.1 MIDlet

MIDlet není přímo knihovnou, ale jedná se o výslednou J2ME aplikaci složenou z knihoven, které proberu dále.

3.1.2 CLDC - Connected Limited Device Configuration

CLDC je jednou z konfigurací J2ME. Je to API určené pro zařízení s omezenou velikostí paměti a spotřebou energie, mezi které patří mobilní telefony.

3.1.3 MIDP – Mobile Information Device Profile

MIDP je profil postavený nad konfigurací CLDC. Existuje ve dvou verzích MIDP 1.0 a MIDP 2.0. MIDP 1.0 je starší verze, která je omezená nižší pamětí, menším rozlišením displeje a dalšími, než novější verze MIDP 2.0. Tato verze je doplněná o mnoho nových tříd a má pro to i větší nároky na zařízení.

3.1.4 MMAPI – Mobile media API

Knihovna MMAPI umožňuje J2ME aplikacím pracovat s multimédií, například zvukem a videem.

3.2 Základní třídy

Zde uvedu nejčastěji využívané třídy. Jelikož pro profesionální aplikaci je využito velké množství tříd, seznámíme se pouze s malým výčtem tříd.

3.2.1 javax.microedition.midlet

Hlavní třída aplikace je vždy rozšířená třídou midlet, která je spojnicí mezi programem pro správu aplikací (AMS) v mobilním telefonu a aplikací. Umožňuje AMS vytvořit, spustit, pozastavit a ukončit midlet v telefonu. Nejdůležitější funkce této třídy jsou startApp, pauseApp a destroyApp, které jsou definovány jako abstraktní a nesmí chybět ani pouhá definice.

3.2.2 javax.microedition.lcdui.Display

Informace podávané uživateli je potřeba nějakým způsobem zobrazit. K tomuto slouží třída Display. Každý midlet má unikátní ukazatel na tento objekt. Samotný objekt není schopen zobrazit na displeji mobilního zařízení žádné informace, ale je schopen zprostředkovat výstup třídy odvozené od Displayable. V jakémkoliv čase běhu aplikace je možné nastavit jinou zobrazitelnou třídu.

3.2.3 javax.microedition.lcdui.Displayable

Třída Displayable je objekt, který disponuje možností být umístěn na výstupním zařízení. Samozřejmě, že tato samostatná třída nepředstavuje žádný výstup, ale poskytuje děděním možnosti potomkům. Přidává možnost definovat příkazy a správce příkazů (Command listener). Příkazy jsou jednotlivé možnosti, které může uživatel přímo ovlivnit volbou na svém mobilním telefonu.

3.2.4 javax.microedition.lcdui.CommandListener

CommandListener není třídou, ale rozhraní určené pro příjem a následné zpracování požadavků uživatele, které má možnost zadávat pomocí navigačních prvků svého mobilního telefonu.

3.2.5 javax.microedition.lcdui.Screen

Všechny zobrazitelné prvky jsou podtřídou Screen. Ta samotná je odvozena z třídy Displayable.

3.2.6 javax.microedition.lcdui.Form

Formulář je jednou z nejpoužívanějších podtříd Screen. Samotný formulář je prázdná oblast, která může obsahovat kromě textu i tzv. předměty (Item) a obrázky. Zaručuje, že všechny prvky vytváření korektní výstup,

3.2.7 Canvas

Profesionální aplikace a Java hry se neobejdou bez grafického rozhraní. K jeho vytvoření se používají většinou třídy odvozené od Canvas. Samostatný Canvas je schopen zprostředkovat grafický výstup.

3.2.8 Manager

Třída Manager umí vytvořit instanci třídy Player ze vstupního proudu nebo na základě zadaného URI.

3.2.9 Player

Rozhraní Player slouží k přehrávání zvukových dat. Během svého životního cyklu se může nacházet v několika z pěti stavů (nerealizovaný, realizovaný, připravený, běžící a ukončený stav.

3.3 Prostředky potřebné pro vývoj aplikací

Zde se zaměřím na prostředky, které jsou nutné pro vývoj aplikací. Mezi tyto prostředky, řadíme vývojové prostředí, emulátory a v poslední řadě mobilní telefony.

3.3.1 Vývojové prostředí

Eclipse je open source vývojová platforma, která je pro většinu lidí známa jako vývojové prostředí (IDE) určené pro programování v jazyce Java. Flexibilní návrh této platformy dovoluje rozšířit seznam podporovaných programovacích jazyků za pomoci pluginů, například o C++ nebo PHP. Právě pluginy umožňují toto vývojové prostředí rozšířit například o návrh UML, či zápis HTML nebo XML.

3.3.2 Emulátory

Vyvíjené aplikace je potřeba otestovat na počítači. K tomuto slouží emulátory. Nejznámější jsou od výrobců NOKIA a SUN. Pokud je aplikace vyvíjená pro více značek mobilních telefonů, je potřeba tuto aplikaci otestovat v co nejvíce emulátorech. Tyto emulátory jsou volně stažitelné na internetu.

3.3.3 Mobilní telefony

Jelikož aplikace se můžou na emulátoru jevit jako plně funkční, nemusí to být ve všech případech pravda. Proto je potřeba svou aplikaci otestovat na jakémkoliv telefonu podporující Javu.



Obrázek 3: Mobilní telefon s dotykovým displejem



Obrázek 4: Mobilní telefon bez dotykového displeje

4 Zadání úkolů

4.1 Konfigurace vývojového prostředí Eclipse pro J2ME

Jedním z prvních úkolů mi byla zadána úplná konfigurace všeho potřebného pro vývoj aplikací v J2ME. Konfigurace spočívá v nastavení vývojového prostředí Eclipse, ve kterém budou aplikace napsány, instalací Sun Wireless Toolkit, což je sada užitečných nástrojů a zdrojových kódů pro J2ME vývoj. V neposlední řadě bylo mým úkolem stáhnout a nainstalovat emulátor pro telefony NOKIA.

4.2 První Midlet pro seznámení s J2ME

Vytvořte první Midlet kde se seznámíte se základními komponentami a jejími využitím. Tento midlet by měl ve výsledku umět využít všechny základní a nejpoužívanější komponenty se základní funkčností. Dále tento Midlet zabalte do užitečné podoby pro instalaci. Poté tento midlet nainstalovat a odzkoušet jeho funkčnost v emulátoru a následně i v mobilním telefonu.

4.3 Vytvořte Midlet MMAPIPlayer pro přehrávání všech podporujících multimediálních souborů

Vytvořte aplikaci, která bude schopná přehrát všechny multimediální soubory, to znamená, že aplikace bude umožňovat přehrát audio a video soubory. Tuto aplikaci naimplementujte pro všechny vhodné zobrazitelné třídy. Jelikož tuto aplikaci budou využívat programátoři pro údržbu platformy, na které pracují, je vhodné zvolit určitý postup plánování, jak bude výsledná aplikace vypadat a komunikaci při vývoji s budoucími uživateli této aplikace. Aplikaci před předáním programátorům odzkoušejte v emulátoru a v mobilním telefonu.

4.4 Rozšíření Midletu MMAPIPlayer o další funkčnost

Doplňte aplikaci o prohlížeč souborů, ve kterém bude možnost prohlížet a vybrat multimediální soubory v základním souborovém systému mobilního telefonu. Přidejte do aplikace ovládání hlasitosti aktuálně přehrávaných souborů. Hodnotu hlasitosti zobrazte na displeji. Aplikaci před předáním programátorům odzkoušejte v emulátoru a v mobilním telefonu.

4.5 Podepsání Midletu MMAPIPlayer

Jelikož midlet bude přistupovat k souborům v paměti telefonu, je potřeba tento midlet podepsat certifikátem a přidat vhodné pravidla. Aplikaci před předáním programátorům odzkoušejte v emulátoru a v mobilním telefonu.

5 Řešení zadaných úkolů

5.1 Konfigurace vývojového prostředí

5.1.1 Konfigurace Eclipse

Pro vytváření J2ME aplikací mi bylo navrženo prostředí Eclipse, protože je to nejčastěji využívaný nástroj pro vývoj aplikací v Javě. Eclipse je volně stažitelná aplikace z internetu zadal jsem klíčové slovo Eclipse do webového vyhledávače. Nástroj Eclipse jsem stáhnul a nainstaloval na PC.

Jelikož nástroj Eclipse v základní instalaci neobsahuje zásuvný modul pro vývoj aplikací v J2ME je potřeba tento plugin doinstalovat do stávající instalace. Postup pro instalaci je následující:

Ve spuštěném nástroji jsem zvolil v hlavním menu volbu Help->Software Updates->Find and Install. Pro další postup nabídne Eclipse vhodného průvodce. V průvodci jsem vybral volbu Search for new features to install a v dalším okně volbu New Remote Site. Do okna jsem vyplnil název a adresu (<http://www.eclipse.org/updates/>), ze které se bude stahovat zásuvný modul pro J2ME. Dále jsem pokračoval podle průvodce, který následně zásuvný modul stáhnul a nainstaloval do Eclipse. Po restartování nástroje byl zásuvný modul importován v Eclipse. J2ME je možno doinstalovat bez připojení k internetu pomocí J2ME paginu staženého v PC. Podrobný postup jak takto plugin doinstalovat je k nalezení na oficiálních stránkách nástroje Eclipse.

5.1.2 Konfigurace Sun Wireless Toolkit

Dalším důležitým nástrojem je Sun Wireless Toolkit, ve kterém je obsažen emulátor od společnosti Sun. Tento emulátor je po přidání do Eclipse výchozím pro spouštění J2ME aplikací. Tento nástroj je také volně stažitelný z internetu. Po stažení jsem nástroj nainstaloval s typickým nastavením instalačního průvodce.

Dále je potřeba tento emulátor přidat do Eclipse. To jsem udělal tak, že při vytvoření prvního projektu J2ME mi Eclipse nabídnul volbu Mange Devices. Po zvolení této volby jsem do Eclipse nabídnutého okna importoval složku kde jsem Sun Wireless Toolkit nainstaloval tedy c:\WTK2.5.2_01. Toto nastavení si Eclipse zapamatoval a nebylo tedy nutné při dalším vytváření projektu J2ME nic importovat.

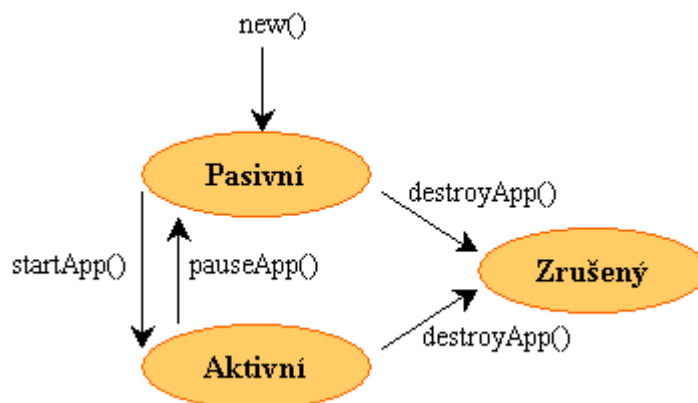
5.1.3 Emulátor NOKIA

Jelikož aplikace jsem měl vyvíjet pro programátory spravující S60 platformu, bylo nutné stáhnout si Emulátor pro telefony NOKIA. Tento nástroj je volně stažitelný na oficiální stránce fóra NOKIE. Po stáhnutí a instalaci není potřeba nic konfigurovat a emulátor je spustitelný z nabídky Start.

5.2 Vytvoření prvního Midletu

5.2.1 Seznámení se s principem fungování Midletu

Pro vytvoření prvního midletu jsem musel nejdřív prostudovat podstatu fungování midletu. Základní třída vždy musí rozšiřovat abstraktní třídu `javax.microedition.midlet.MIDlet`. Při spuštění aplikace je tato třída vytvořena zavoláním svého veřejného konstruktoru bez parametrů. Midlet se může nacházet v několika stavech, jak je vysvětleno na obrázku 3. Běh aplikace a přechod mezi stavy řídí aplikační

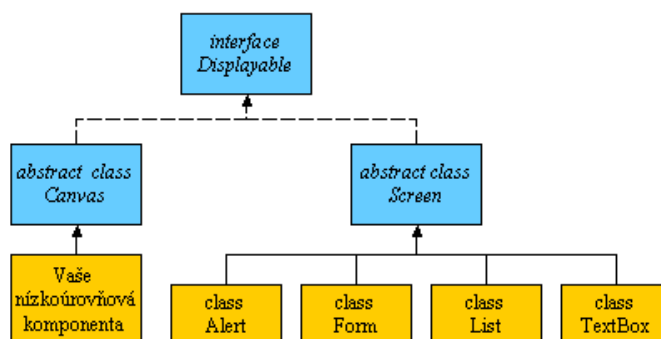


Obrázek 3: Život midletu

manažer. Aplikace se po zavolání konstruktoru nachází ve stavu pasivním. Po zavolání metody `startApp()` přechází midlet do aktivního stavu kde inicializuje zdroje typu `Thread` a `Connection`. Metoda `pauseApp()` se vrací aplikace do pasivního stavu a měla by tyto zdroje uvolnit. Aplikace je ukončena aplikačním manažerem zavoláním metody `destroyApp()`.

5.2.2 Rozhodnutí o použití komponent pro základní funkce Midletu

V prvním midletu jsem se rozhodl použít některé komponenty, které jdou zobrazit na displeji. Na obrázku 4 je zobrazena jejich hierarchie. Dále jsem se rozhodl použít příkazy, což jsou nejdůležitější prvky, kterými se midlet ovládá. Pomocí Canvasu



Obrázek 4 Komponenty zobrazitelné na displeji

jsem vytvořil vlastní odvozenou třídu, ve které jsem využil základní vykreslovací prvky.

5.2.3 Vlastní řešení prvního projektu

Ve vývojovém prostředí Eclipse jsem vytvořil nový projekt J2ME Midlet Suite, což jsou projekty pro vytváření midletů. První projekt jsem pojmenoval first_Midlet. V projektu bylo potřeba vytvořit první základní třídu Midlet vybráním volby J2ME Midlet a první Midlet jsem pojmenoval Test_Midlet. Eclipse sám do této třídy vytvořil základní tři metody startApp(), pauseApp() a destroyApp().

Do třídy Midlet bylo potřeba importovat balík tříd lcdui (javax.microedition.lcdui.*), ve kterém jsou všechny potřebné třídy. Třidu Midletu jsem rozšířil o rozhraní CommandListener, které se stará o obsluhu příkazu přiřazených midletu. Eclipse doplnil do midletu neimplementovanou metodu rozhraní CommandAction, která se stará obsluhu příkazů pro tento midlet. Dále jsem vytvořil čtyři nové třídy myForm, myList, myTextBox a myCanvas, které dědí z tříd Form, List, TextBox a Canvas. Do všech tříd jsem opět importoval známý balík tříd lcdui. Tyto třídy mají také mimo jiných parametrů konstruktory, parametr midlet, jelikož se z těchto tříd budou volat metody hlavní třídy midletu.

V hlavní třídě midletu bylo potřeba vytvořit objekty těchto nových tříd, protože se s nimi bude dále pracovat. Také jsem zde vytvořil objekt typu Form který jsem pojmenoval firstForm a bude představovat jednoduchou úvodní obrazovku. Do hlavní třídy jsem také dodal příkazy a následně je přiřadil objektu firstForm. Objektu firstForm musel být také nastaven posluchač, aby příkazy odchytil a vyvolával příslušné funkce midletu. Příkazy jsou objekty třídy Command. Vytvořil jsem tedy tyto příkazy: cmdToForm, cmdToList, cmdToTextBox, cmdToCanvas a cmdExit. První čtyři příkazy budou zajišťovat přepínání do mnou vytvořených tříd a příkaz cmdExit midlet ukončí. Třídy myForm, myCanvas, myList a myTextBox jsem také doplnil o příkaz cmdBack a také jsem všem nastavil posluchače. Příkaz cmdBack bude pouze volat metodu midletu pro navrácení na úvodní obrazovku firstForm.

Třída myForm byla doplněna o zobrazení jednoduchého textu, který se při přepnutí obrazovky objeví na displeji.

Ve třídě myList jsem vytvořil čtyři položky listu, mezi kterými je možnost přepínat.

Třídě myTextBox jsem nastavil počáteční text, který se zobrazí při přepnutí, a také jsem omezil jeho velikost na 256 znaků, které bude možno do textového pole napsat.

Poslední třídu myCanvas jsem vyplnil žlutou barvou a přidal jednoduchý text. Pro psaní textu ve třídě Canvas je zapotřebí užít třídy Font, která umožňuje nastavit různé vlastnosti textu, jako je jeho velikost, barva atd.

5.2.4 Zabalení, instalace a spuštění Midletu

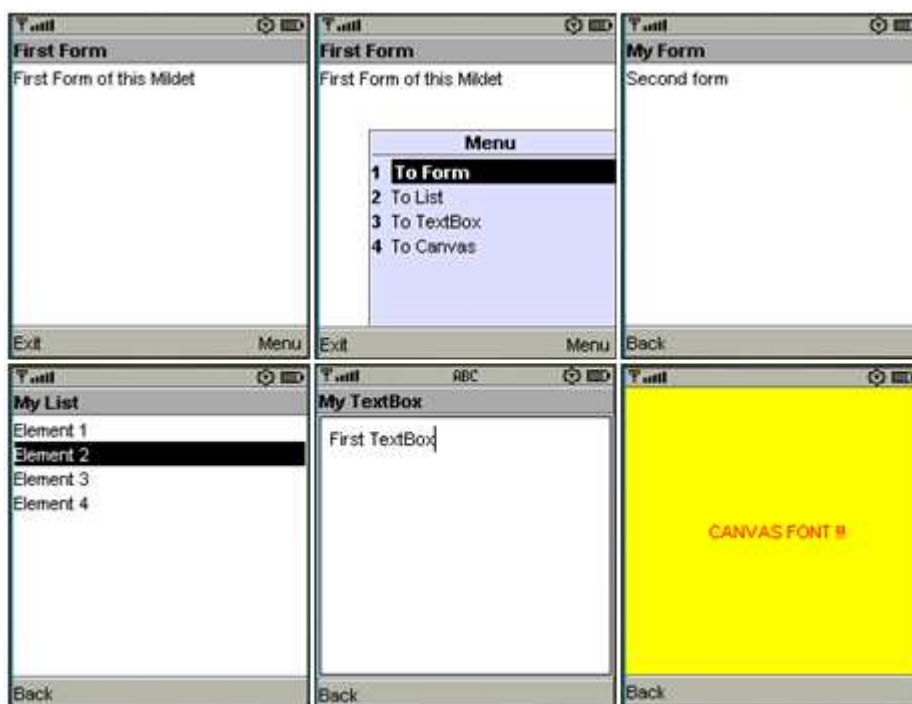
Námí vytvořený midlet je potřeba zabalit do podoby vhodné pro instalaci do emulátoru nebo mobilního telefonu. Vývojové prostředí s doplněným zásuvným

modulem umožňuje pomoci volby CreatePackage tento midlet zabalit do souboru s příponou jar, která je příznačná pro Java soubory. Dále je také vytvořen soubor s příponou jad, který obsahuje různé informace o midletu jako název, velikost atd. Také se zde přidává různá pravidla a certifikát při podepisování midletu.

Midlet se může spustit přímo v Eclipse, kde je nastaven emulátor od Sun. Jednou z dalších možností je emulátor spustit ze start nabídky a vybrat jad soubor pro spuštění. Pro spuštění midletu v emulátoru NOKIA je potřeba tento midlet nejdříve nainstalovat a pak spustit. Instalace se provede tak, že midlet se nakopíruje do složky epoc32\wincsw\c\Data\Installs\ v místě, kde je emulátor Nokia nainstalován. Poté už stačí pomocí správce souboru v emulátoru midlet vybrat, nainstalovat a spustit. Postup pro mobilní telefon je velmi podobný. Liší se pouze tím, že aplikaci je potřeba do telefonu nahrát pomocí bluetooth nebo USB kabelu. Dále stačí postupovat stejně jako v emulátoru.

5.2.5 Obrazovky midletu

Na obrázku 5 jsou vyobrazeny všechny obrazovky midletu.



Obrázek 5 Obrazovky prvního midletu: (zleva) první formulář, první formulář s výběrovým menu, druhý formulář, seznam se čtyřmi položkami, textové pole a vlastní Canvas s textem

5.3 Vytvoření midletu MMAPi

5.3.1 Plánování

Jelikož tento midlet budou používat programátoři pro údržbu S60 platformy, musí se rozhodnout, co všechno bude výsledný midlet obsahovat a které části jsou zanedbatelné a nemusí být tedy implementovány.

Na počátku každého implementačního úkolu se musí provést plánování úkolu. Plánování se provádí tak, že všichni programátoři, kteří budou s výslednou aplikací pracovat, jsou pozváni na meeting, kde se probírá funkčnost výsledné aplikace. Na základě získaných informací jsem se rozhodnul, které komponenty použiji. Každý den se pak provádí tak zvané scrum meetingy, kde se schází celý programátorský tým a probírá se, co každý programátor za uplynulý den udělal. V průběhu celé implementace jsem musel komunikovat s programátory a doplňovat výslednou funkčnost, která nebyla zjištěna počátku plánování.

5.3.2 Zvolené komponenty pro MMAPi midlet

Základní třídou pro MMAPi je třída Manager (`javax.microedition.media.Manager`). Pomocí této třídy je pak možno vytvořit instanci třídy Player, která slouží pro přehrávání zvukových a video souborů. Pro zobrazení videa jsem se rozhodl použít třídu Form a také třídu Canvas. Midlet bude obsahovat i jednoduché příkazy pro ovládání midletu.

5.3.3 Vlastní řešení MMAPi midletu

Tento projekt jsem pojmenoval MMAPi_Midlet a do něho jsem vytvořil základní třídu Midlet, kterou jsem pojmenoval Test_mmapi. Do projektu jsem vložil také jeden hudební a jeden zvukový soubor, které bude možno dále přehrávat v midletu. Dále jsem vytvořil dvě další třídy myCanvas a myForm pro zobrazení videa.

V základní třídě midletu bylo potřeba vytvořit objekt typu Form, který je výchozí obrazovkou pro tento midlet a bude obsahovat základní příkazy pro ovládání midletu. Objekt typu Form jsem pojmenoval first_form a přiřadil jsem mu posluchače pro obsluhu příkazů. Vytvořil jsem také těchto pět příkazů: `cmdPlaySonForm`, `cmdPlayVonForm`, `cmdPlaySonCanvas`, `cmdPlayVonCanvas` a `cmdExit`. Tyto příkazy budou obsluhovat spuštění videa respektive zvuku ve třídě myForm nebo myCanvas. Příkaz `cmdExit` midlet ukončí. Třída first_form bude také obsahovat základní text pro úvodní obrazovku.

Do vlastních tříd myCanvas a myForm bylo potřeba importovat základní balík tříd `javax.microedition.lcdui.*` a také `javax.microedition.media.*`, která obsahuje důležité třídy pro veškerou práci s multimediálními soubory.

Třída myForm dědí z třídy Form a může tedy využívat všechny jeho funkce. Této třídě jsem přiřadil posluchače pro obsluhu příkazů a také jsem přidal jeden příkaz `cmdBack`, pomocí kterého se midlet bude vracet na úvodní obrazovku. Do této třídy bylo potřeba také vytvořit vstupní proudy, což jsou objekty typu `InputStream`, kterým se přiřadí vložené multimediální soubory. Pomocí metody `createPlayer` třídy Manager jsem vytvořil novou instanci třídy Player jak pro zvukový tak video soubor. Parametrem třídy Player je pak vstupní proud, kterým se Player sváže se souborem a také typ souboru který se bude přehrávat. Pro zvukový soubor pak stačí zavolat metodu `start()` třídy Player. Poté multimediální soubor začne hrát. Pro video soubor je, ale potřeba vytvořit objekt typu `GUIControl`, který je schopen zobrazit video ve třídě Form. Třídě `GUIControl` se přiřadí

Player, který bude video přehrávat. Poté jsem opět zavolal metodu start() a na displeji se objevil přehrávaný video soubor.

Třída myCanvas dědí z třídy Canvas a postup pro přehrávání je zcela stejný jako pro třídu myForm. Jediný rozdíl, v kterém se liší způsob implementace, je ten, že namísto objektu třídy GUIControl je použit objekt typu VideoControl. Proto jsem tedy vytvořil objekt typu VideoControl a přiřadil jsem mu Player v kterém se bude přehrávat video soubor. Dále jsem ve zavolal metodu setVisible objektu VideoControl, která zařídí zobrazení videa ve třídě myCanvas. Poté jsem zavolal metodu start() objektu Player a video se zobrazilo na displeji ve třídě my Canvas.

5.3.4 Odzkoušení a předání Midletu

Midlet jsem podle známého postupu nainstaloval do emulátoru a také do telefonu. Ověřil jsem jeho funkčnost v obou zařízeních a předal jej programátorům pro jejich práci. Jelikož s aplikací budou pracovat pouze programátoři, nebyla potřeba vytvářet jakákoliv dokumentace.

5.4 Rozšíření midletu MMAPI

5.4.1 Plánování

MMAPI midlet bylo potřeba rozšířit o jeho další funkčnost. Po diskuzi s programátory jsem zjistil, že do midletu musím přidat jednoduchý správce souborů, protože doposud midlet uměl pouštět určité typy souborů, které byly do midletu vloženy. Po rozšíření o správce souborů bude tedy možno vybrat jakýkoliv multimediální soubor. Dále bylo potřeba do midletu doplnit ovládání hlasitosti a hodnotu hlasitosti zobrazit na displeji. Na základě zjištěných poznatků jsem se rozhodl, jaké použiji komponenty.

5.4.2 Zvolené komponenty pro rozšíření midletu

Pro rozšíření jsem se rozhodl užít již hotové dvě třídy myForm a myCanvas. Dále bylo potřeba zajistit zobrazení stromu souborů v telefonu, proto jsem se rozhodl pro třídu List, pomocí které bude následně soubory vybírat. Další důležitou třídou je FileConnection, která se stará o práci se soubory. Pro ovládání hlasitosti bylo zapotřebí použít třídu VolumeControl, pomocí které je možno nastavit přehrávači hlasitost právě přehrávaného souboru. Opět jsem také využil třídy Command pro obsluhu midletu.

5.4.3 Vlastní rozšíření midletu MMAPI

Vytvořil jsem nový projekt, který jsem nazval TotalCommander a do něho jsem vytvořil midlet, který jsem také nazval TotalCommander. Dále jsem do projektu importoval vytvořené třídy myCanvas a myForm.

Pro správce souborů jsem se rozhodl vytvořit novou třídu, kterou jsem pojmenoval FileCommander. Tato třída dědí z třídy List. Do této třídy jsem importoval známý balík tříd javax.microedition.lcdui.* a také javax.microedition.io.file, který zajišťuje práci se soubory. Třídě FileCommander jsem přiřadil posluchače pro obsluhu příkazů a přidal jsem příkazy na otvírání složek repektive na otvírání souborů. Také bylo

nutno vytvořit příkaz pro návrat z adresářů a na ukončení midletu. Příkaz pro otevření souborů dále nabídnul otevření souboru buďto ve třídě myForm a nebo myCanvas. Do třídy FileCommander se po spuštění midletu vždy vloží seznam adresářů, které telefon obsahuje. Při otevření složky se celý seznam smaže a vytvoří se nový seznam, který obsahuje dané podadresáře otevřeného adresáře. Po návratu zpět opět seznam vymaže a vypíše seznam starý. Při otevření konkrétního souboru jsem si uložil celou cestu k tomuto souboru a tu následně předal zobrazovacím třídám myForm a my Canvas.

Třídy myForm a myCanvas bylo potřeba rozšířit o metodu, která převezme z třídy FileCommander cestu k souboru a následně ji předá přehrávači, aby vybraný soubor přehrál. Příkaz cmdBack jsem v obou třídách upravil tak aby se vrátil zpět do třídy FileCommander. Dále jsem v obou třídách vytvořil nový objekt typu VolumeControl, který jsem následně přiřadil objektu Player na obsluhu hlasitosti. Hodnotu hlasitosti jsem zobrazil na displej.

5.4.4 Odzkoušení a předání midletu

Výslednou funkčnost midletu jsem opět odzkoušel v emulátoru a následně pak v telefonu. Tento midlet jsem předal programátorům, kteří ho využívají při údržbě S60 platformy.

5.5 Podepsání midletu

5.5.1 Podepsání midletu TotalCommander

Vytvořený midlet pracuje se soubory na paměti telefonu a tedy telefonu musíme pokaždé povolit přístup k vybranému souboru. Aby tedy telefon aplikaci nezastavoval a ptal se na přístup k souborům je potřeba tuto aplikaci podepsat. Po té se již telefon nebude dotazovat na přístup k souborům.

Aplikace jsem podepsal pomocí souboru s příponou *.ks (Keystore File), který mi byl přidělen a používá se vždy při podepisování aplikací. V Eclipse jsem zvolil vlastnosti aplikace, kde jsem mně přiřazený soubor vložil a vyplnil hesla potřebné k užití tohoto souboru. Po té jsem aplikaci pomocí Eclipse zabalil do jad a jar souboru. V jad souboru už byl přidán nutný certifikát pro midlet. Do jad souboru jsem přidal pravidla potřebné pro přístup k souborům v telefonu.

5.5.2 Odzkoušení midletu

Midlet jsem opět nainstaloval do emulátoru a telefonu. Po spuštění aplikace jsem již nemusel povolit přístup k souborům, ale vše zajišťoval přidáný certifikát. Výslednou aplikaci jsem předal programátorům.

6 Závěr

Zde bych chtěl zhodnotit průběh mé praxe ve firmě TietoEnator. Jsem rád, že jsem si namísto běžné bakalářské práce zvolil právě praxi. Mohl jsem se tedy seznámit s mladým kolektivem lidí, kteří zde každodenně vykonávají svou práci.

Bylo umožněno nahlédnout do nových technologií, o kterých jsem doposud neměl ani tušení. Byl jsem také seznámen s pracovními postupy, které zaměstnanci využívají denně. Jsem rád, že mí kolegové vycházeli všem mým požadavkům vstřícně a byli mi vždy nápomocni. Já jsem se snažil nenarušovat jejich běžný pracovní systém a plnit úkoly, které mi zadávali a konzultovali se mnou.

V průběhu praxe jsem získal mnoho praktických zkušeností a rad do budoucna.

7. Literatura

1. Beginning J2ME Platform

From Novice to Professional, Third Edition, Sing Li and Jonathan Knudsen,
Copyright 2005

2. J2ME (online)

aktualizováno 15. 4. 2008, Dostupný na : <http://interval.cz/vyvoj-aplikaci/j2me/>

3. Eclipse vývojové prostředí

aktualizováno 28. 4. 2009, Dostupný na :

[http://cs.wikipedia.org/wiki/Eclipse_\(v%C3%BDvojov%C3%A9_prost%C5%99ed%C3%AD\)](http://cs.wikipedia.org/wiki/Eclipse_(v%C3%BDvojov%C3%A9_prost%C5%99ed%C3%AD))